**Falco安装**

# curl -s https://s3.amazonaws.com/download.draios.com/DRAIOS-GPG-KEY.public | apt-key add -

# curl -s -o /etc/apt/sources.list.d/draios.list <https://s3.amazonaws.com/download.draios.com/stable/deb/draios.list>

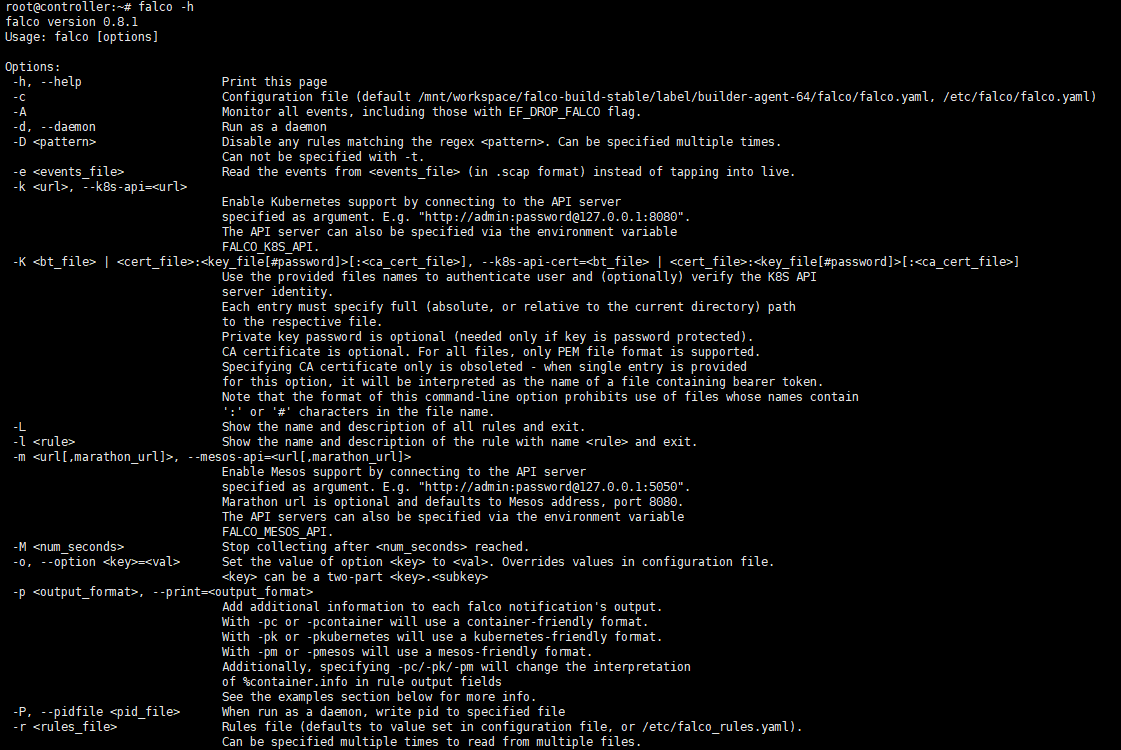
(PS : draios.list文件名根据不同的环境进行相应更改)

# apt-get update

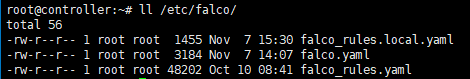
# apt-get -y install falco

**Falco命令**

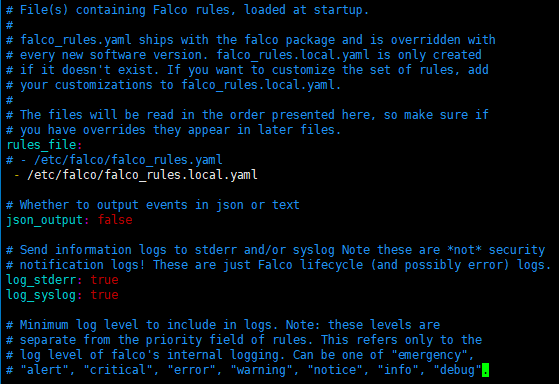
# falco –h



Falco是基于策略集来监控用户行为的，而这些策略集定义在yaml文件中：



falco.yaml定义常规配置以及使用哪些策略文件：

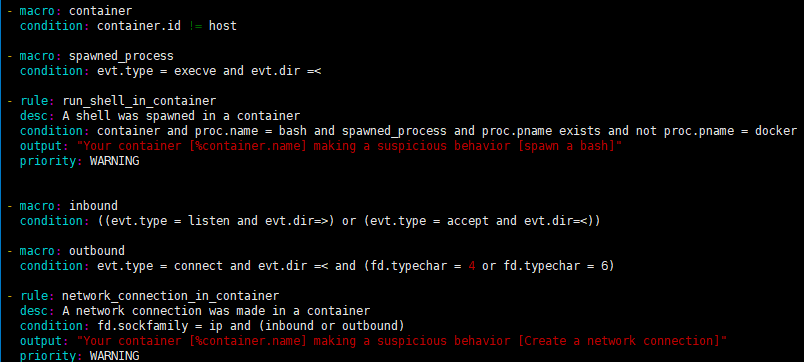


falco\_rules.yaml是Falco官方提供的策略文件，里面定义了一些诸如敏感文件操作、异常socket连接、创建namespace等恶意行为的警告策略

（PS：默认使用，但这里为了方便测试我将它注释了，只读取falco\_rules.local.yaml的策略）

接下来我们编辑falco\_rules.local.yaml：

vi /etc/falco/falco\_rules.local.yaml



- macro: container

condition: container.id != host

宏定义一个container(container.id不是host)

- macro: spawned\_process

condition: evt.type = execve and evt.dir =<

宏定义了一个spawned\_process(开始调用execve)

- rule: run\_shell\_in\_container

desc: A shell was spawned in a container

condition: container and proc.name = bash and spawned\_process and proc.pname exists and not proc.pname = docker

output: "Your container [%container.name] making a suspicious behavior [spawn a bash]"

priority: WARNING

rule：策略名

desc：描述

condition：策略，container宏，进程名bash，spawned\_process宏，父进程存在，父进程名不是docker

output：输出

priority：等级

- macro: inbound

condition: ((evt.type = listen and evt.dir=>) or (evt.type = accept and evt.dir=<))

宏定义一个inbound(开始调用listen()或结束调用accept())

- macro: outbound

condition: evt.type = connect and evt.dir =< and (fd.typechar = 4 or fd.typechar = 6)

宏定义一个outbound(结束调用connect()，IPv4或IPv6)

- rule: network\_connection\_in\_container

desc: A network connection was made in a container

condition: fd.sockfamily = ip and (inbound or outbound)

output: "Your container [%container.name] making a suspicious behavior [Create a network connection]"

priority: WARNING

rule：策略名

desc：描述

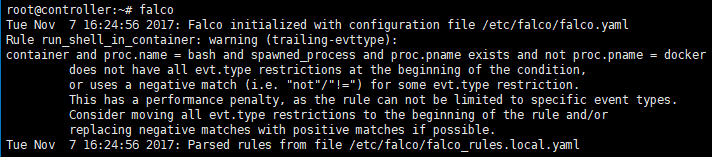
condition：策略，sock类型ip，宏inbound，宏outbound

output：输出

priority：等级

策略语法手册：<https://www.sysdig.org/wiki/sysdig-user-guide/>

# falco 启动falco

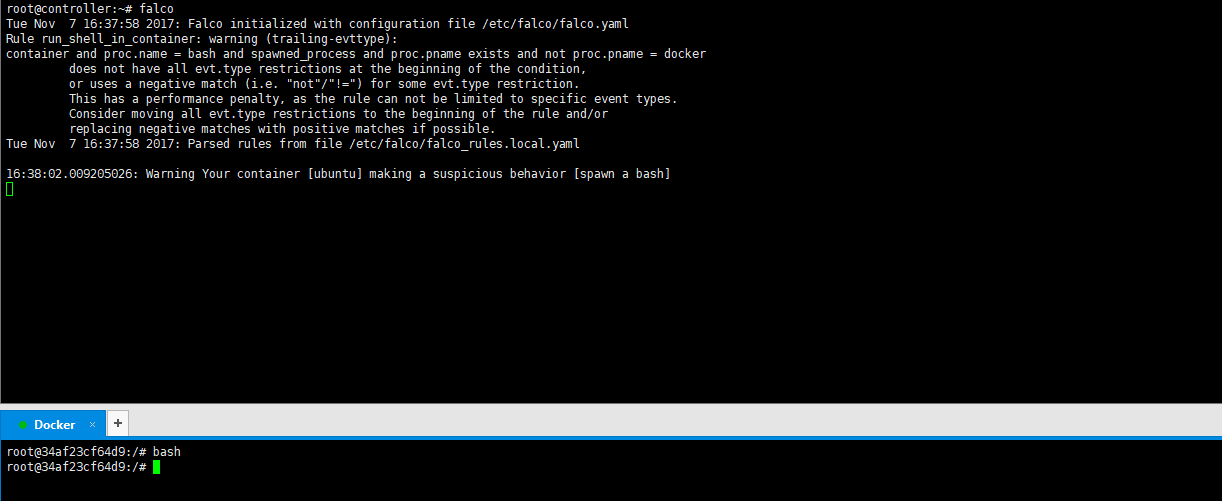


提示你的策略没有涵盖所有情况，我们现在只是测试，忽略此警告

现在我们启动一个ubuntu容器：

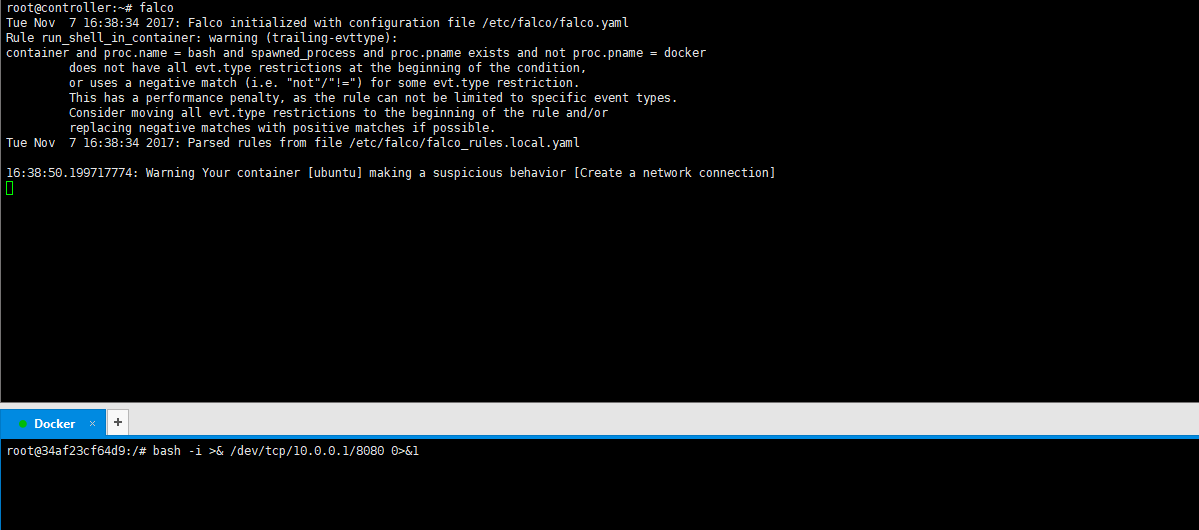


在ubuntu容器中新启一个bash：



falco检测到了ubuntu容器的spawn bash行为，并发出警告

在ubuntu容器中启一个反弹shell：



当容器接收到C&C服务器的连接时Falco会发出警告